Day - 8

(Class- Movie)

Question:

The class Movie is stated below. An instance of class Movie represents a film. This class has the following three properties:

- title, which is a String representing the title of the movie
- studio, which is a String representing the studio that made the movie
- rating, which is a String representing the rating of the movie (i.e. PG13, R, etc)

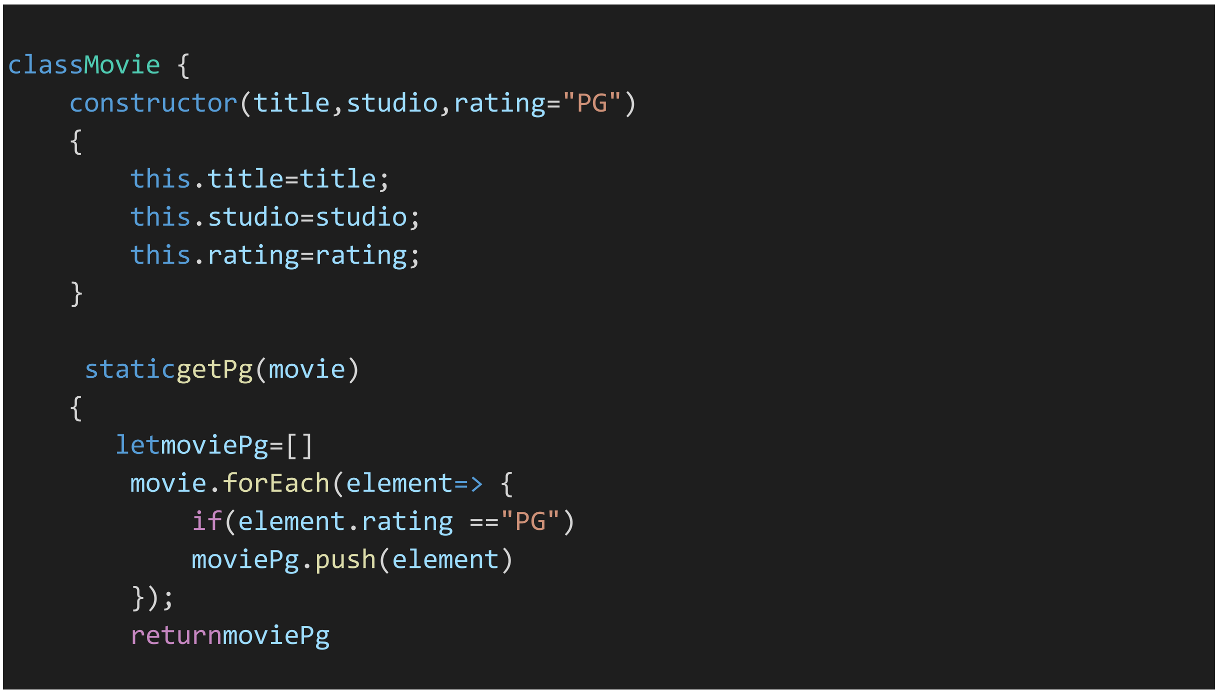
a) Write a constructor for the class Movie, which takes a String representing the title of the movie, a String representing the studio, and a String representing the rating as its arguments, and sets the respective class properties to these values.

b) The constructor for the class Movie will set the class property rating to "PG" as default when no rating is provided.

c) Write a method getPG, which takes an array of base type Movie as its argument, and returns a new array of only those movies in the input array with a rating of "PG". You may assume the input array is full of Movie instances. The returned array need not be full.

d) Write a piece of code that creates an instance of the class Movie with the title “Casino Royale”, the studio “Eon Productions”, and the rating “PG13”

Answer:



```
classMovie {
    constructor(title,studio,rating="PG")
    {
        this.title=title;
        this.studio=studio;
        this.rating=rating;
    }

     staticgetPg(movie)
    {
       letmoviePg=[]
        movie.forEach(element=> {
            if(element.rating =="PG")
            moviePg.push(element)
        });
        returnmoviePg
    }
}
letnewMovie1=newMovie("Casino Royale","Eon Productions","PG-13");
letnewMovie2=newMovie("Beast","SunPictures","PG");
letnewMovie3=newMovie("Doctor","AGS Entertainment",);
letarr=[newMovie1,newMovie2,newMovie3];
letPGMovies=Movie.getPg(arr);
console.log(PGMovies);
```

Output:

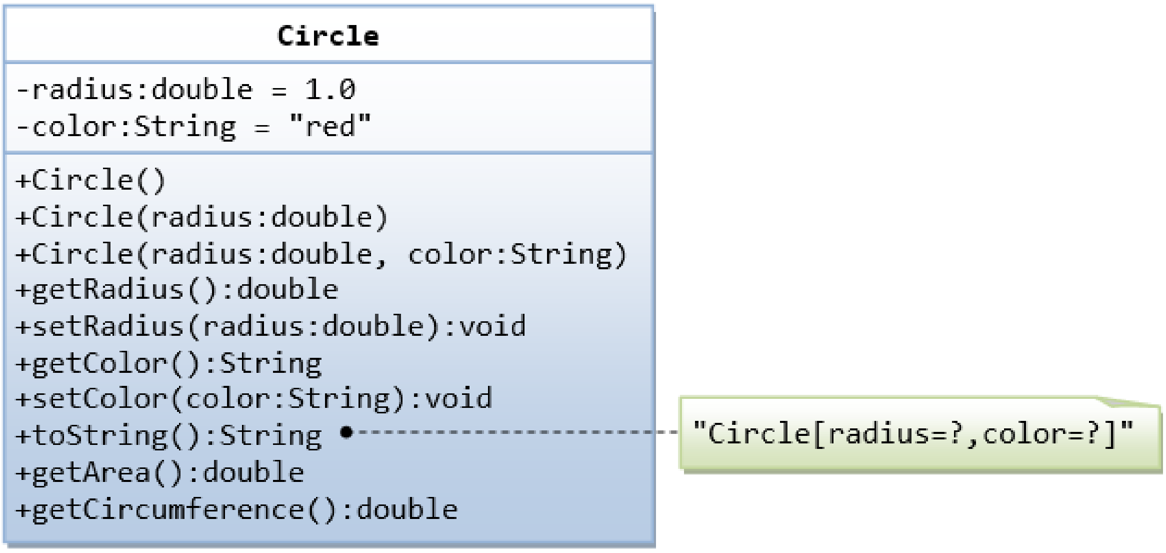
[Movie, Movie]

```
0: Movie {title: 'Beast', studio: 'Sun Pictures', rating: 'PG'}
1: Movie {title: 'Doctor', studio: 'AGS Entertainment', rating: 'PG'}
length: 2
[[Prototype]]: Array(0)
```

Circle - Class

2. Question:

Convert the UML diagram to Typescript class. - use number for double



```
classCircle{
    constructor(radius,color)
    {
        this._radius=radius;
        this._color=color;
    }

    getradius(){
        returnthis._radius
    }
    setradius(value){
        this._radius =value

    }
    getcolor(){
        returnthis._color
    }
    setcolor(value){
        this._color=value
    }

    getArea(){
        returnMath.PI*Math.pow(this._radius,2)
    }

    getCircum(){
        return2*Math.PI*this._radius
    }

}
letcircle1= newCircle(1,"Yellow")

console.log(circle1.getArea())
console.log(circle1.getCircum())
```

Output:

```
3.141592653589793

6.283185307179586
```

3. Write a “person” class to hold all the details.

```
classPerson{
    constructor (firstName,lastName,age){
        this._firstName=firstName;
        this._lastName=lastName;
        this._age=age;
    }

    getDetails(){
        return [this._firstName,this._lastName,this._age]
    }
}

letmyData= newPerson("Ninja","Narseh",26)

console.log(myData.getDetails())
```

Output:

['Ninja', 'Naresh', 26]

4. Write a class to calculate uber price.

```
classUber{
    constructor(totalKm,waitingTime=0)
    {
        this.totalKm=totalKm;
        this.waitingTime=waitingTime
    }
    calculateFare(){
        if(this.totalKm<=2)
        {
        return (30 + ' Rs')
        }

        this.totalKm =this.totalKm-2
        return (30+(this.totalKm*4)+(this.waitingTime*2)+' Rs')


    }
}

letRide1 =newUber(2)
console.log(Ride1.calculateFare())

letRide2 = newUber(82,5)
console.log(Ride2.calculateFare())
```

Output:

30 Rs

360 Rs